AI Eyes - Smart Surveillance System

A comprehensive AI-powered surveillance system with real-time object detection, face recognition, and suspicious activity monitoring.

🎯 Features

- **Multi-Camera Support**: Automatic detection and monitoring of IP webcam cameras
- **YOLOv9 Object Detection**: Real-time detection of persons, weapons, and objects
- **Face Recognition**: LBPH-based authorized vs intruder identification
- **Activity Analysis**: Loitering, crowd detection, abandoned objects, weapon alerts
- **Live Web Dashboard**: Unified monitoring interface for all cameras
- **Real-time Alerts**: Instant notifications for suspicious activities

🚀 Quick Start

Prerequisites

- Python 3.8+
- IP webcam cameras accessible on your network
- Modern web browser

Installation

1. **Clone the repository**

```
git clone https://github.com/Praveen9964935712/AI-Eyes-on-Security.git
cd AI-Eyes-on-Security
```

2. **Install dependencies**

```
cd backend
pip install -r requirements.txt
```

3. **Start the surveillance system**

```
python multi_camera_surveillance.py
```

4. **Access the dashboard** Open your browser to `http://localhost:5002`

📁 Project Structure

```
AI-Eyes-on-Security/
├── backend/                    # Backend surveillance system
│   ├── surveillance/           # AI surveillance modules
│   │   ├── detector.py         # YOLOv9 object detection
│   │   ├── face_recognition.py # LBPH face recognition
│   │   ├── activity_analyzer.py # Suspicious activity detection
│   │   ├── surveillance_manager.py # Main coordinator
│   │   └── alert_manager.py    # Alert and notification system
│   ├── app/                    # Flask application
│   ├── config/                 # Configuration files
│   ├── database/               # Database models
│   ├── storage/                # File storage management
│   ├── multi_camera_surveillance.py # Main application
│   ├── live_surveillance_system.py  # Single camera system
│   └── requirements.txt        # Python dependencies
├── src/                        # Frontend React application
├── data/                       # Training data and known faces
└── README.md                   # This file
```

🔧 Configuration

Adding Authorized Faces

1. Create folders in `data/known_faces/`
2. Add 3-5 photos per person:

```
data/known_faces/
├── john_doe/
│   ├── photo1.jpg
│   ├── photo2.jpg
│   └── photo3.jpg
└── jane_smith/
    ├── photo1.jpg
    └── photo2.jpg
```

Camera Setup

The system automatically detects IP cameras on your network. Supported formats:

- `http://IP:PORT/video`
- `http://IP:PORT/stream`

🎮 Usage

Multi-Camera Surveillance

```
cd backend
python multi_camera_surveillance.py
```

- Automatically detects all available IP cameras
- Provides unified web dashboard at `http://localhost:5002`
- Start/stop individual cameras or all cameras

Single Camera Surveillance

```
cd backend
python live_surveillance_system.py
```

- Processes single IP camera feed
- Real-time AI detection and activity analysis

Alert Types

- 🔴 **CRITICAL**: Weapon detection, unauthorized access
- 🟡 **WARNING**: Loitering, abandoned objects, crowd detection
- 🔵 **INFO**: Normal monitoring, object detection updates

API Endpoints

- `GET /api/status` - System status and statistics
- `POST /api/start_all` - Start surveillance on all cameras
- `POST /api/stop_all` - Stop all surveillance
- `GET /video_feed/<camera_name>` - Live video stream

System Requirements

- **CPU**: Multi-core processor (Intel i5+ or AMD equivalent)
- **RAM**: 8GB+ recommended for multiple cameras
- **GPU**: CUDA-compatible GPU optional for faster inference
- **Network**: Stable connection to IP cameras
- **Storage**: 10GB+ for logs and snapshots

Security Features

1. **Real-time Object Detection**: YOLOv9 model for accurate detection
2. **Face Recognition**: LBPH algorithm for intruder identification
3. **Activity Monitoring**: Advanced behavioral analysis
4. **Multi-Camera Correlation**: Cross-camera activity tracking
5. **Alert System**: Instant notifications and logging

Contributing

1. Fork the repository
2. Create a feature branch
3. Commit your changes

4. Push to the branch
5. Create a Pull Request

License

This project is licensed under the MIT License - see the LICENSE file for details.

Future Enhancements

- [ ] Mobile app integration
- [ ] Cloud storage support
- [ ] Advanced AI models (YOLOv10, Transformer-based)
- [ ] Voice alerts
- [ ] Integration with security systems
- [ ] Advanced analytics dashboard

Support

For issues and questions:

- Create an issue on GitHub
- Check the documentation
- Review configuration settings

AI Eyes - Keeping watch with artificial intelligence